



## Mobile Agents for Database Applications

Ludwig Klug



Ludwig Klug

- Ludwig works with Agents for more than 1.5 years
- Technical consultant for Grasshopper implementations
- Implementation of DBagents



## Grasshopper

### Building an Agent with Grasshopper

## Grasshopper – The Agent Platform

### ■ Producer

- IKV++ GmbH, Berlin, Germany
- Partner of GMD FOKUS

### ■ Costs

- Evaluation free of charge
- Licenses ~10000 € (for 100 VAR licenses)
- 3000 € / Year for premium support (answers within 24 hrs)
- Discussion forum (answers usually within 1 week)

### ■ FIPA and MASIF Extensions

- One of few agent platforms building on standards
- Extensions realized as Grasshopper agents

### ■ External Security

- Protect agents against external threats
- SSL (e.g., IAIK Implementation)

### ■ Internal Security

- Protect platforms against hostile agents
- Protect agents against hostile platforms
- Public key encryption (certificates)

### ■ Authorization

- Restrict Access to Agency Resources
  - Directories
  - Sockets
- Policy Files (for Each Place Separate)
  - Current version only allows the same access rights for all agents
  - Next version: codesource-based access control

### ■ Missing: Protecting agents against hostile platforms

- Agent-to-Agent, Agent-to-Platform, Agent-to-Registry
- Communication Over Proxies
- Synchronous Communication
  - Wait for a return value
- Asynchronous Communication
  - Do not wait for a return value
- Broadcasts
  - Communicate with more object simultaneously

### ■ Region

- Unit of agencies
- Region registry maintains agent addresses

### ■ Agency

- Container for places
- Usually one agency on one computer

### ■ Place

- Runtime environment for agents

### ■ Agent

- Code that runs as a thread in a place
- (Optional) can move between places

### ■ What is an agent?

- Instance of subclass of MobileAgent or StationaryAgent
- Has a unique identifier (IP-address, time, copy-number)

### ■ MobileAgent

- Allowed to move
- PersistentMobileAgent

### ■ StationaryAgent

- Limited to a single place
- PersistentStationaryAgent

## ■ **public void init(Object[] args)**

- Initialization of the agent (agent parameters)
- Called after the agent is loaded

## ■ **public void live()**

- Heart of the agent (agent's active behaviour)
- Only method that **MUST** be overridden

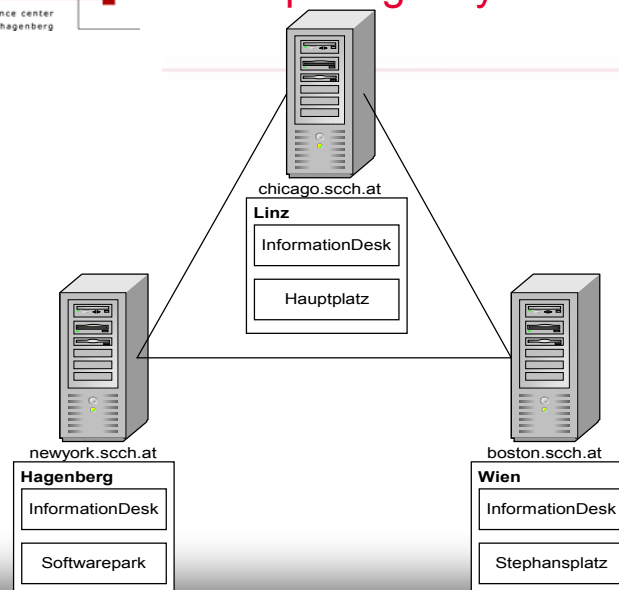
## ■ **public void move (GrasshopperAddress location)**

- Migrate to another location

## ■ **public void remove()**

- Destroy an agent.

- before/afterMove
  - Called before/after the agent migrates
- before/afterCopy
  - Called before/after the agent is cloned
- beforeRemove
  - Called before the agent is removed

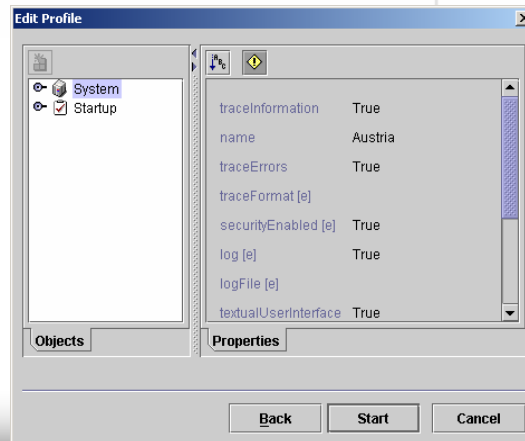




## Creating a Region Registry

### Decide on Which Computer It Should Run

- Registry is vital for agencies
- Computer should run permanently

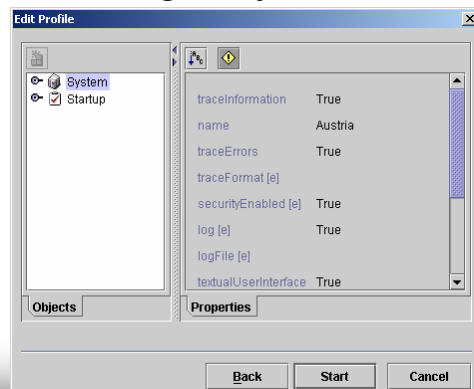


## Creating a Region Registry

### Three Ways To (Configure and) Run an Agency

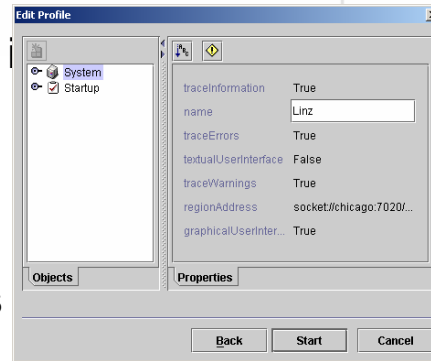
- Commandline argument
- Wizard
- Profile editing

### GUI/TUI option



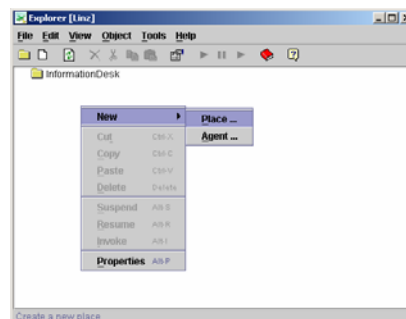
## Creating the Agencies

- Name of the Agency
- Decide Which Region it Should Belong to
- Access Control
- Security
- Enter Region Address
  - Format:  
socket://chicago:7020/Austria
  - protocol://host:port/region



## Creating Places

- Place InformationDesk is Predefined
  - Place where every agent can migrate to
  - No security limits

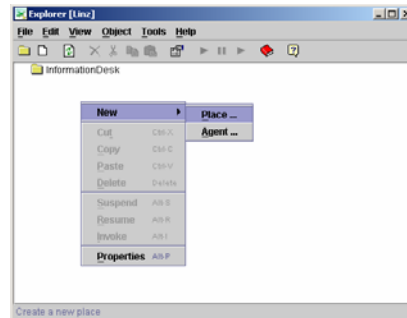


Place creation in an agency

## Creating Places

### ■ Create Custom Places

- Specify a name
- Specify security policy



Place creation in an agency

## ■ 3 Ways of Running an Agent

### ■ GUI

- In an agency explorer by right-clicking and selecting "New Agent" on a place's popup menu

### ■ Profile

- The user can define an agent in the configuration window, which is started on initialization of the agency

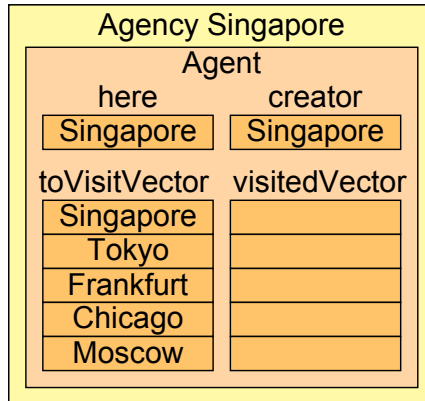
## 3 Ways of Running an Agent

### ■ API

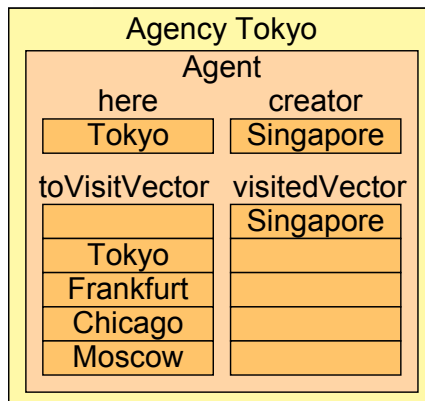
- During execution of an agent another agent can be activated by calling the createAgent-method of the Grasshopper API

## Agent life cycle

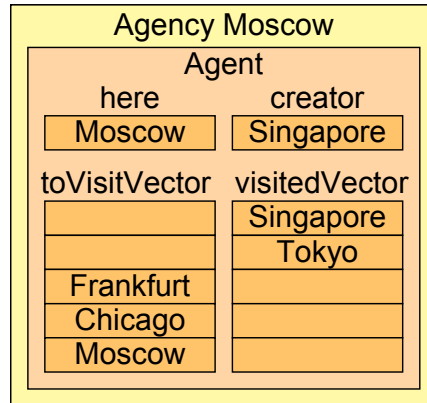




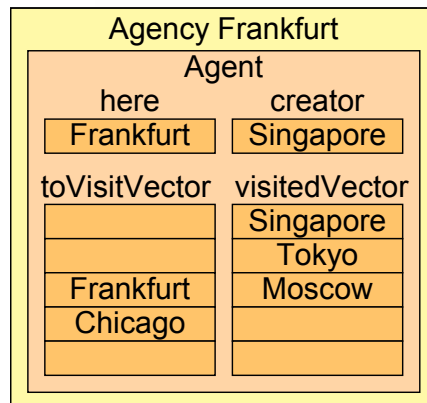
Agent.move(Tokyo)



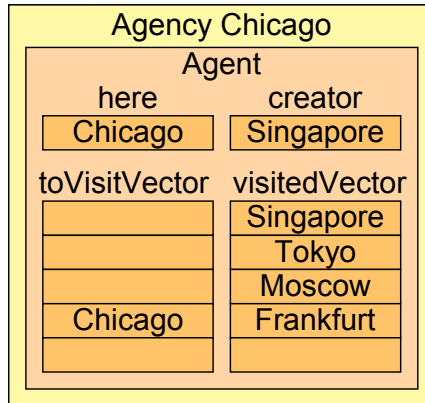
Agent.move(Moscow)



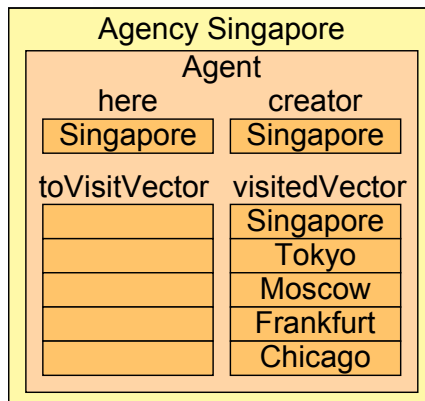
Agent.move(Frankfurt)



Agent.move(Chicago)



Agent.move(Singapore)





## DBagents

### Agents in Database Systems



## DBagents

- Basic platform runtime required on each computer

### Advantages

- Database queries within DBMS
- Searching within network
- Load balancing (not between DB servers but processing at the DB where data is located)



■ Standard Agent Platform for Transportation

- Extend available migration strategies
- Extend available security solutions
- Perform non-SQL operations (e.g., schema export)
- Transport query results back to home agency

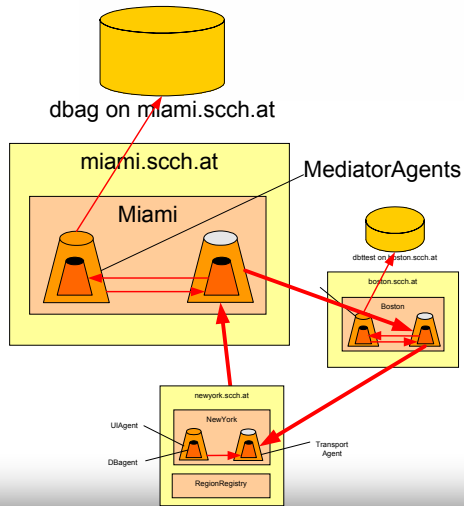


■ Mobile Database Agents

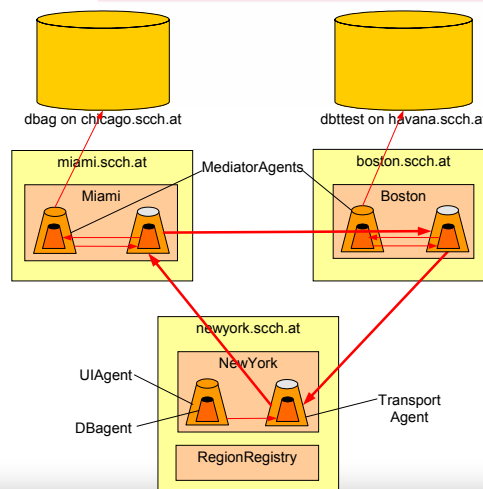
- SQL statements
- SQL scripts
- Java stored procedure



## Implemented Architecture



## Implemented Architecture



DBagents provides a **flexible API**  
to  
support agent-based  
**operations for DBs.**

- **4 Agent types**
- **4 Result handlers**
- **2 Communication modes**
- **Agent groups** support multi-agent transactions
- **UI templates** for results, logs and migration strategies

### ■ SQL

- A statement is executed and results are returned to home

### ■ SQL script

- A script is executed and log information is returned

### ■ Stored procedure installation

### ■ Java stored procedure installation

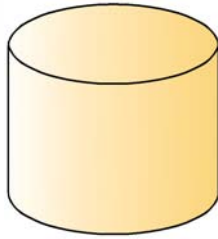
### ■ Status

- Basic prototype implemented
- Currently, test scenarios are being developed
- Implementation of CopyAgent in Progress
- Problem for real-world applications:
  - Stability
  - Security



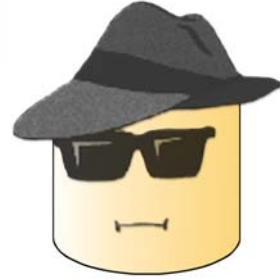
Mobile  
Agents

+



Distributed  
Databases

=



DBagents